



مروری بر کاربرد برنامه متلب در حل محاسبات عددی ریاضی

پوهنیار احمدولی نوری^۱، پوهنمل فرهاد حکیمی^۲، پوهنیار محمد یزدان پرست^۳

دیارتمنت ریاضی، پوهنځی تعلیم و تربیه پوهنتون فراه

ایمیل: ahmadwalinoori@farau.edu.af

چکیده

در عصر حاضر پیچیدگی مسائل علمی و تحقیقی نیاز به ابزارهای محاسباتی دقیق، منعطف و کارآمد را به صورت چشمگیری افزایش داده است. متلب (MATLAB-Matrix Laboratory) به عنوان یک برنامه محاسباتی پیشرفته و زبان برنامه نویسی سطح بالا، جایگاهی برتری در میان همچون برنامه‌ها کسب کرده است. این تحقیق با هدف بررسی بر کاربردهای کلیدی برنامه متلب به حل محاسبات عددی ریاضی در الجبر خطی، حل معادلات خطی و غیرخطی، مشتق، انتگرال، معادلات تفاضلی، ماتریکس‌ها و تحلیل‌های احصائیوی می‌پردازد. چارچوب نظری این تحقیق مبتنی بر روش کتابخانه‌ای بوده و با بررسی منابع معتبر علمی سعی شده است تا قابلیت‌های برنامه متلب در حل مسائل پیچیده محاسباتی مورد بررسی قرار گیرد. یافته‌ها نشان می‌دهد که متلب با برخورداری از ابزارهای تخصصی، توابع از پیش تعریف شده، قابلیت تجسم‌سازی پیشرفته و پشتیبانی از زبان‌های برنامه نویسی شی گراء، محیط توانمند برای انجام تحلیل‌های پیچیده محاسبات ریاضی، یادگیری و پیاده سازی الگوریتم‌ها و اجرای پروژه‌های بین رشته‌ای فراهم می‌سازد. در حقیقت، می‌توان گفت که متلب ابزاری قابل اعتماد، انعطاف پذیر و تحول آفرین در حل محاسبات پیچده ریاضی به شمار می‌رود.

کلمات کلیدی: کاربرد متلب، مسائل ریاضی، محاسبات عددی



علم ریاضیات، دانش اساسی با کاربردهای گسترده در صنایع متنوع و حوزه‌های علمی گوناگون است. در طول زمان ریاضیدانان برای حل محاسبات پیچیده ریاضی به ابزارهای محاسباتی متکی بوده‌اند که برنامه MATLAB یکی از این ابزارها است که به‌طور وسیعی در تحقیق‌های ریاضی مورد استفاده قرار گرفته است. MATLAB مخفف (Matrix Laboratory) بوده، که نخستین بار در اوایل دهه ۱۹۸۰ میلادی توسط دانشمند کلیو مولر (Cleve Moler)، ریاضیدان آمریکایی، برای تسهیل محاسبات ماتریکسی توسعه یافت (Moler & Little, 2020). این برنامه یک محیط انعطاف‌پذیر برای محاسبات عددی، تحلیل داده‌ها و تجسم‌سازی فراهم می‌آورد و اهمیت و موثریت آن در شاخه‌های مختلف ریاضی و توانایی آن را برای تحول تحقیق‌های ریاضی برجسته می‌سازد (Pawade, 2023). این برنامه دارای کتابخانه جامع، توابع داخلی، جعبه ابزارها و دستورات کلیدی بوده و به محققان این امکان را می‌دهد تا به‌سرعت نمونه‌سازی اولیه الگوریتم‌ها را انجام دهند، داده‌ها را تحلیل نموده و نتایج را بصری‌سازی نمایند. برنامه MATLAB همچنین یک محیط برنامه‌نویسی منعطف فراهم می‌کند که محققین را قادر می‌سازد تا الگوریتم‌های اختصاصی خود را پیاده‌سازی کرده و الگوریتم‌های موجود را خصوصی‌سازی نمایند. یکی از مزایای کلیدی استفاده از MATLAB در حل مسائل ریاضی، توانایی آن در مدیریت مجموعه داده‌های بزرگ و انجام محاسبات پیچیده است. این امر آن‌را به یک انتخاب رایج در حوزه‌هایی مانند حل الگوریتم‌های ریاضی، الگوریتم‌های یادگیری ماشین، الگوریتم‌های یادگیری عمیق و محاسبات هوشمند تبدیل کرده است، جایی که محققین اغلب با حجم وسیعی از داده‌ها سروکار دارند (Chen, 2022; Pawade, 2023).

ابزارهای قدرتمند تجسم‌سازی برنامه MATLAB همچنین آن را به یک ابزار ارزشمند برای محققین تبدیل نموده است که نیاز به تجسم داده‌ها و انتقال نتایج



تحقیقی خود دارند. MATLAB می‌تواند گراف‌های دو بعدی و سه بعدی با کیفیت بالا و همچنین بصری‌سازی و تجسم‌سازی‌های تعاملی تولید کند، که می‌توان از آن‌ها برای انتقال مفاهیم پیچیده ریاضی به مخاطبان گسترده‌تر استفاده کرد. افزون بر این جامعه کاربری فعال MATLAB و مستندات جامع آن، یادگیری این زبان و دریافت رهنمایی در صورت نیاز را برای محققین تسهیل می‌سازد. این برنامه همچنین انبوهی از ابزارها و کدهای ایجاد شده توسط کاربران را فراهم می‌کند که می‌توان از آن‌ها برای بهینه‌سازی جریان کار تحقیق و ارتقای بهره‌وری استفاده نمود. در نهایت، MATLAB به دلیل انعطاف‌پذیری، توانمندی و سهولت استفاده، یک ابزار ارزشمند برای تحقیق در علوم ریاضیات محسوب می‌گردد. این ابزار می‌تواند به محققان کمک کند تا به سرعت نمونه‌سازی و پیاده‌سازی الگوریتم‌ها را انجام دهند، مجموعه داده‌های بزرگ را تحلیل کرده و نتایج را به صورت بصری به گونه‌ای که برای مخاطبان گسترده‌تر قابل درک باشد، ارائه نمایند (Pawade, 2023)

هدف اصلی این مقاله بررسی، تحلیل جامع بر کاربردهای برنامه MATLAB در حل محاسبات ریاضی است. این تحقیق با بررسی گسترده مطالعات و تحقیقات پیشین، ضمن بررسی قابلیت‌ها و مزایای MATLAB در تسهیل حل محاسبات ریاضی پیچیده، تجزیه و تحلیل داده‌های بزرگ و ارائه بصری نتایج و محدودیت‌های موجود در به‌کارگیری این ابزار را نیز شناسایی می‌کند. علاوه بر این، این مقاله چشم‌انداز نوین و فرصت‌های آینده استفاده از MATLAB را در توسعه علوم ریاضی مورد بحث قرار می‌دهد تا زمینه‌ای مستحکم برای تحقیقات آتی فراهم آیند.

روش تحقیق

این تحقیق از نوع مروری - کتابخانه‌ای بوده و با اتکاء به منابع معتبر علمی شامل مقالات علمی - تحقیقی و کتب تخصصی از پایگاه‌های معتبر علمی



فراه علمی-خبرنیز ژورنال

جمع آوری شده است. هدف از این تحقیق، ارائه تحلیلی جامع، ساختارمند و مستند از کاربردهای MATLAB در حل محاسبات ریاضی است.

کاربرد MATLAB در الجبر خطی

الجبر خطی یکی از شاخه‌های اساسی و زیربنایی علوم ریاضیات به‌شمار می‌رود که به بررسی معادلات خطی و روش‌های تحلیلی جهت یافتن پاسخ آن‌ها می‌پردازد. برنامه MATLAB با بهره‌مندی از کتابخانه‌ای جامع از توابع عددی و سمبولیک (symbolic)، بستری توانمند و دقیق برای انجام محاسبات پیشرفته در این حوزه فراهم می‌سازد. به‌عنوان نمونه، از عملگر بک‌اسلش (1) جهت حل سیستم‌های معادلات خطی استفاده می‌شود که روشی مؤثر برای دستیابی به پاسخ‌های دقیق و بهتر فراهم می‌آورد. همچنین، توابع اختصاصی برای محاسبه قیمت مشخصه (Eigenvalues) و وکتورهای مشخصه (Eigenvectors) امکان تحلیل ساختاری ماتریکس‌ها و حل مسائل طیفی را فراهم می‌سازد. علاوه بر این، قابلیت‌های تعبیه‌شده برای انجام ضرب ماتریکس‌ها و معکوس‌سازی ماتریکس‌ها، موجب شده است تا برنامه MATLAB به‌عنوان ابزاری استاندارد و قابل اعتماد در تحلیل‌های پیچیده الجبر خطی در محیط نهادهای تحصیلی و تحقیقی شناخته شود (Xue, 2020).

توابع کاربردی در برنامه MATLAB

برنامه MATLAB به‌عنوان یکی از ابزارهای قدرتمند محاسبات عددی و فنی، مجموعه‌ای گسترده از توابع ریاضی از پیش تعریف‌شده را در اختیار کاربران قرار می‌دهد. این توابع شامل عملیات ابتدایی و پیشرفته ریاضی بوده و در حوزه‌های متنوعی مانند ریاضی، فزیک، اقتصاد و تحلیل داده کاربرد دارند. برای دسترسی کامل به فهرست توابع ابتدایی (Elementary Functions) و توابع خاص (Special Functions) می‌توان از دستورات `help elfun` و `help specfun` در محیط MATLAB استفاده نمود. این دستورات اطلاعات جامعی درباره نحوه استفاده از



این توابع و کاربرد آن‌ها ارائه می‌دهند. جدول ۱ فهرستی از توابع پرکاربرد ریاضی در برنامه MATLAB را نشان می‌دهد. نکته قابل توجه آن است که این توابع می‌توانند بر روی اعداد اسکالر، وکتورها و ماتریکس‌ها اعمال شوند. برنامه MATLAB با بهره‌گیری از این توابع، امکان انجام محاسبات پیچیده و تحلیل‌های عددی را به سادگی و با دقت بالا فراهم می‌سازد (Geete & Verma, 2023).

جدول ۱: توابع اساسی و کاربردی در برنامه MATLAB

$\cos(x)$	کوساین	$abs(x)$	قیمت مطلق
$\sin(x)$	ساین	$sign(x)$	تابع علامت (<i>Signum</i>)
$\tan(x)$	تانجانت	$max(x)$	حداکثر
$acos(x)$	ارک کوساین (معکوس کوساین)	$min(x)$	حداقل
$asin(x)$	ارک ساین (معکوس ساین)	$ceil(x)$	رونداف کردن به طرف مثبت بی‌نهایت
$atan(x)$	ارک تانجانت (معکوس تانجانت)	$floor(x)$	رونداف کردن به طرف منفی بی‌نهایت
$exp(x)$	تابع نمایی	$round(x)$	رونداف کردن به نزدیک‌ترین عدد صحیح
\sqrt{x}	جذر	$rem(x)$	باقیمانده تقسیم
$\log(x)$	لوگاریتم طبیعی	$angle(x)$	زاویه
$\log_{10}(x)$	لوگاریتم معمولی (قاعده ۱۰)	$conj(x)$	مزدوج مختلط

حل معادلات خطی در MATLAB

حل معادلات خطی یکی از مسائل اساسی در ریاضیات و علوم ساینسی به‌شمار می‌رود که در بسیاری از کاربردهای علمی و عملی مطرح می‌گردد. در



محیط برنامه MATLAB، دستور solve یکی از پرکاربردترین ابزارها برای حل معادلات خطی به‌شمار می‌رود. این دستور قادر است معادلات را نسبت به متحول یا متحول‌های مشخص شده به‌صورت تحلیلی حل نماید. برای استفاده از این دستور، ابتدا باید متحول‌های مورد نظر به‌صورت سمبولیک تعریف شوند. دستور solve تنها در حل معادلات تکی، بلکه در حل سیستم‌های چندمعادله‌ای خطی نیز عملکردی دقیق و مؤثر دارد. این دستور قادر است سیستم‌های با دو یا چند مجهول را حل کرده و نتایج را به‌صورت عددی نمایش دهد. خروجی معمولاً به‌صورت ساختاری یا ویکتوری از مقادیر متحول‌ها ارائه می‌گردد.

از مزایای مهم این دستور، سادگی در کاربرد، دقت بالا و توانایی در تحلیل ساختار معادلات خطی است. به‌کمک دستور solve می‌توان به‌سرعت پاسخ سیستم معادلات را به‌دست آورد، بدون آن‌که نیاز به نوشتن روش‌های عددی مانند حذف گاوسی یا تجزیه ماتریکسی باشد. در نهایت دستور solve یک ابزار کلیدی در MATLAB برای حل دقیق و تحلیلی معادلات خطی می‌باشد. (Lopez, 2014).

مثال ۱: حل معادله خطی یک مجهوله در برنامه MATLAB

$$2x + 5 = 11$$

```
Command Window
>> syms x
eq = 2*x + 5 == 11;
sol = solve(eq, x);
disp(sol)
3
```

شکل ۱: حل معادله یک مجهوله در برنامه MATLAB

مثال ۲: حل معادله خطی دو مجهوله در برنامه MATLAB

$$\begin{cases} 2x + y = 7 \\ x - y = 1 \end{cases}$$



```

Command Window
>> syms x y
eq1 = 2*x + y == 7;
eq2 = x - y == 1;
sol = solve([eq1, eq2], [x, y]);
disp(sol.x)
8/3
5/3

```

شکل ۲: حل معادله دو مجهوله در برنامه MATLAB

مثال ۳: حل معادله خطی سه مجهوله در برنامه MATLAB

$$\begin{cases} x + 2y - z = 4 \\ 3x - y + 2z = 5 \\ 2x + y + z = 6 \end{cases}$$

```

Command Window
>> syms x y z
eq1 = x + 2*y - z == 4;
eq2 = 3*x - y + 2*z == 5;
eq3 = 2*x + y + z == 6;
sol = solve([eq1, eq2, eq3], [x, y, z]);
disp(sol.x)
disp(sol.y)
disp(sol.z)
3/2
11/6
7/6

```

شکل ۳: حل معادله سه مجهوله در برنامه MATLAB

حل معادلات غیر خطی در MATLAB

در برنامه MATLAB، زمانی که با سیستم معادلات غیرخطی روبه‌رو هستیم و هدف یافتن پاسخ عددی برای آن است، از دستور `vpsolve` استفاده می‌کنیم. این دستور برای حل معادلاتی طراحی شده است که به دلیل پیچیدگی یا ساختار غیرخطی شان، نمی‌توان آن‌ها را به صورت سمبولیک حل کرد یا به روش‌های تحلیلی



دستور solve که در یافتن پاسخ قادر نیستند با استفاده از روش‌های عددی و تقریبی، جواب‌هایی با دقت بالا (اعشاری) ارائه می‌دهد. ساختار نوشتاری این دستور بسیار مشابه solve است؛ تنها تفاوت اصلی در نوع خروجی آن است، که به صورت عددی و سمبولیک نشان داده می‌شود. برای استفاده از این دستور، کافی است معادلات مورد نظر و متحول‌های مجهول را مشخص کنیم. در مواردی که چندین جواب ممکن وجود دارد یا معادلات شامل توابع غیرخطی مانند توان، جذر، لوگاریتم یا مثلثاتی باشند، vpasolve راه‌حلی دقیق و مؤثر برای رسیدن به جواب‌های ممکن فراهم می‌سازد. این قابلیت MATLAB، به‌خصوص در حل محاسبات پیچیده‌ی ریاضیات کاربردی و علوم محاسباتی بسیار ارزشمند است (Norddin et al., 2025).

مثال ۱: حل سیستم معادلات غیر خطی دو مجهوله

$$\begin{cases} y^3 + y \sin(2x) = 1 \\ y^2 + 1 - e^{-\frac{x}{3}} = 0 \end{cases}$$

```

Command Window

eq1 = y^3 + y*sin(2*x) - 1 == 0;
eq2 = y^2 + 1 - exp(-x/3) == 0;

sol = vpasolve([eq1, eq2], [x, y]);

if isempty(sol)
    disp('مجموع جواب این به‌دقت نشد.')
else
    x_rounded = round(double(sol.x)*100)/100;
    y_rounded = round(double(sol.y)*100)/100;

    disp('مخاز x:')
    disp(x_rounded)
    disp('مخاز y:')
    disp(y_rounded)
end

مخاز x:
-1.7400
مخاز y:
0.8900
    
```

شکل ۴: حل معادله غیر خطی دو مجهوله در برنامه MATLAB

عملیات روی ماتریکس‌ها در برنامه MATLAB (Matrix Operations)
 برنامه MATLAB مخفف (Matrix Laboratory) است، دلیل نام‌گذاری آن این است که ساختار اصلی برنامه MATLAB بر اساس ماتریکس‌ها طراحی شده



است؛ به طوری که حتی یک عدد ساده نیز به صورت یک ماتریکس (1×1) در نظر گرفته می‌شود. بیشتر محاسبات در MATLAB با استفاده از عملیات ماتریکسی انجام می‌شود. این عملیات شامل اعمال ساده‌ای مانند جمع (+)، تفریق (-)، ضرب (*) و تقسیم (/) و همچنین ضرب عنصر به عنصر (*.) است. برای مثال، دستور $A * B$ ضرب ماتریکس معمولی را انجام می‌دهد، در حالی که $A .* B$ عناصر متناظر دو ماتریکس را در هم ضرب می‌کند.

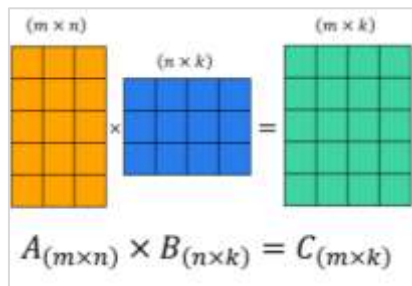
علاوه بر عملیات اساسی، توابع پیشرفته‌ای مانند محاسبه معکوس ماتریکس (inv)، دترمینانت (det)، جابجای ماتریکس (')، حل سیستم معادلات خطی (A \ B) و محاسبه مقادیر و وکتور خاص (eig) را نیز در اختیار کاربران قرار می‌دهد. این توابع، حل مسائل پیچیده‌ی ریاضی را ساده و سریع می‌سازند. یکی از مزیت‌های مهم MATLAB، امکان انجام محاسبات ویکتوری (Vectorized Computing) است؛ به این معنا که بدون نیاز به حلقه‌های تکراری، می‌توان یک عملیات را بر روی کل یک ماتریکس یا وکتور انجام داد. این ویژگی باعث افزایش سرعت اجرا و کاهش پیچیدگی برنامه‌نویسی می‌شود. در نسخه‌های جدید MATLAB، همچنین امکاناتی برای پردازش موازی و استفاده از کارت گرافیک (Graphics Processing Unit) فراهم شده است که اجرای سریع‌تر محاسبات سنگین، مانند تحلیل داده‌های بزرگ و مدل‌سازی در یادگیری ماشین را ممکن می‌سازد. در مجموع، آشنایی با عملیات ماتریکسی در MATLAB، پایه‌ای ضروری برای استفاده مؤثر از این برنامه است و نقش مهمی در تحلیل‌های عددی و مدل‌سازی علمی ایفاء می‌کند (Thabet & Al-Ghuribi, 2012).

ضرب ماتریکس‌ها در MATLAB

ضرب ماتریکس‌ها یکی از عملیات کلیدی در الجبر خطی است. هدف از ضرب ماتریکس‌ها، ترکیب اطلاعات دو ماتریکس به گونه‌ای است که روابط میان



داده‌ها حفظ شده و قابل تحلیل باشد. در شکل ۵، ضرب دو ماتریکس به صورت تصویری نمایش داده شده است. ماتریکس اول A دارای ابعاد $(n \times m)$ است؛ به این معنا که شامل m سطر و n ستون می‌باشد. ماتریکس دوم B دارای ابعاد $(k \times n)$ است؛ یعنی دارای n سطر و k ستون می‌باشد. شرط اصلی برای انجام ضرب ماتریکس‌ها این است که تعداد ستون‌های ماتریکس اول با تعداد سطرهای ماتریکس دوم برابر باشد تنها در این صورت است که عمل ضرب ممکن و معتبر خواهد بود (Overman, 2011).



شکل ۵: قانون ضرب ماتریکس، ترکیب ابعاد و ساختار آن

مثال ۱: ضرب ماتریکس $(A \times B)$ که ماتریکس A دارای ۴ سطر و ۳ ستون است و ماتریکس B دارای ۳ سطر و ۲ ستون است در نتیجه ماتریکس C که از حاصل ضرب A و B بدست آمده دارای ۴ سطر و ۲ ستون می‌باشد

```
Command Window
>> A = [1 2 3; 4 5 6; 7 8 9; 2 1 0];
B = [1 2; 0 1; 1 0];
C = A * B

C =

     4     4
    10    13
    16    22
     2     5
```

شکل ۵: عملیه ضرب ماتریکس در برنامه MATLAB

MATLAB در آنالیز حقیقی و انتیگرال



MATLAB ابزاری کارآمد برای حل مسائل پیچیده در آنالیز حقیقی و انتگرال است؛ شاخه‌ای از ریاضیات که به مطالعه‌ی نرخ تغییرات و تجمع می‌پردازد. برای نمونه، MATLAB توابعی مانند quad و diff را فراهم می‌کند که برای انتگرال‌گیری عددی و مشتق‌گیری عددی ضروری هستند. این توابع می‌توانند برای یافتن مساحت زیر یک منحنی یا محاسبه‌ی مشتق یک تابع به کار روند و مفاهیم اساسی در معادلات تفاضلی و انتگرال محسوب می‌شوند. با بهره‌گیری از ابزارهای MATLAB برای انتگرال‌گیری عددی و مشتق‌گیری عددی، محققین و محصلان می‌توانند مسائل آنالیز حقیقی و انتگرال را به‌شکلی کارآمدتر و با دقت بالاتر حل کنند (Petrás, 2011).

محاسبه انتگرال تحلیلی و عددی در MATLAB

انتگرال‌گیری یکی از اساسی‌ترین مفاهیم ریاضیات است که در بسیاری از مسائل فنی، علمی و انجینری کاربرد دارد. برنامه‌ی MATLAB ابزارهای بسیار مناسبی برای محاسبه‌ی انتگرال‌ها فراهم کرده است، به‌طور خاص در مواقعی که محاسبه‌ی دستی دشوار یا زمان‌بر است. در MATLAB، انتگرال‌ها به دو روش اصلی محاسبه می‌شوند: روش سمبولیک و روش عددی.

۱. برای محاسبه‌ی به‌روش سمبولیک، از دستور `int` استفاده می‌شود. ابتدا باید متحول مورد نظر را به‌صورت سمبولیک با دستور `syms` تعریف کرده و سپس تابع را در قالب `int(expr, var, a, b)` به MATLAB معرفی کنید. اگر حدود `a` و `b` مشخص شوند، انتگرال معین محاسبه می‌شود؛ در غیر این صورت، نتیجه به‌صورت انتگرال نامعین خواهد بود.

۲. در صورتی که انتگرال به‌صورت تحلیلی قابل حل نباشد یا بخواهیم صرفاً مقدار عددی آن را داشته باشیم، از دستور `integral` استفاده می‌شود. در این روش تابع باید به‌شکل تابع ناشناس تعریف شده باشد و ساختار آن به صورت `integral(fun, a, b)` است (Pawade, 2023; Petrás, 2011).



فراه علمی-فخرینیز ژورنال
 مثال ۱: در مثال زیر، انتیگرال تابع x را از 0 تا $\sin(t)$ به صورت تحلیلی محاسبه می‌نماید.

$$\int_0^{\sin(t)} x \, dx$$

```
Command Window
>> syms x t
int(x,0,sin(t))

ans =

sin(t)^2/2
```

شکل ۶: حل انتیگرال تحلیلی با برنامه MATLAB

مثال ۲: محاسبه انتیگرال عددی در MATLAB با استفاده از دستور integral

$$\int_0^{\pi} \sin \frac{x}{\sqrt{x+1}} \, dx$$

```
Command Window
>> f = @(x) sin(x./sqrt(x+1));
integral(f,0,pi)

ans =

2.2844
```

شکل ۷: حل انتیگرال عددی در برنامه MATLAB

مشتق‌گیری در MATLAB

مفهوم مشتق یکی از اصول اساسی و کاربردی در ریاضیات و انجینری است که برای بررسی نرخ تغییرات و رفتار توابع به‌کار می‌رود. در بسیاری از کاربردهای علمی، محاسبه مشتق به صورت دستی ممکن است زمان‌بر و همراه با خطا باشد. برنامه MATLAB با استفاده از ابزارهای سمبولیک، این پروسه را ساده و دقیق



می سازد. در MATLAB، دستور diff برای محاسبه مشتق استفاده می شود طوری که متحولها به صورت سمبولیک تعریف شوند و ساختار کلی این دستور به صورت $\text{diff}(F, \text{var}, n)$ نوشته می شود که در آن F تابع مورد نظر، var متحول مشتق گیری و n مرتبه مشتق را مشخص می نماید (Kazimovich & Guvercin, 2012).

مثال ۱: مشتق تابع $f(x) = x^2 + 3x + 2$ نسبت به x به صورت زیر

محاسبه می شود.

$$f'(x) = \frac{d}{dx}(x^2 + 3x + 2)$$

$$= 2x + 3$$

$$f'(x) = 2x + 3$$

```
Command Window
>> syms x
f = x^2 + 3*x + 2;
diff(f)

ans =

2*x + 3
```

شکل ۸: محاسبه مشتق مرتبه اول تابع در MATLAB

کاربرد MATLAB در حل معادلات تفاضلی

معادلات تفاضلی نوعی از معادلات ریاضی هستند که رابطه میان یک تابع و مشتقهای آن را توصیف می کنند. برنامه MATLAB مجموعه ای گسترده از توابع عددی را برای حل معادلات تفاضلی ارائه می دهد که از جمله مهم ترین آنها می توان به توابع ode23 و ode45 اشاره کرد. این توابع ابزارهایی کلیدی در حل معادلات تفاضلی پیچیده ای هستند که به صورت تحلیلی قابل حل نیستند. توانایی



برنامه MATLAB در پردازش مجموعه داده‌های بزرگ، این برنامه را به ابزاری توانمند و مؤثر در تحلیل و حل سیستم‌های پیچیده معادلات تفاضلی تبدیل کرده است. معادلات تفاضلی به دو دسته تقسیم می‌شوند؛ معادلات تفاضلی معمولی (Ordinary Differential Equations - ODE) و معادلات تفاضلی با مشتقات جزئی (Partial Differential Equations - PDE). در معادلات ODE تنها یک متحول مستقل وجود دارد، به طوری که تمامی مشتقات موجود در معادله نسبت به همین متحول گرفته می‌شوند. در مقابل، معادلات PDE شامل مشتقات نسبت به دو یا چند متحول مستقل هستند (Arefin et al., 2020).

حل سیستم معادلات تفاضلی در MATLAB

برای حل یک سیستم معادلات تفاضلی معمولی در MATLAB، می‌توان از دستور `dsolve` استفاده نمود. ساختار کلی این دستور به گونه‌ای است که معادلات مورد نظر به صورت یک آرایه از معادلات وارد می‌شوند؛ به عنوان مثال، `dsolve([eqn1, eqn2])` برای حل دو معادله‌ی همزمان استفاده می‌گردد. نکته مهم در این روش آن است که تعداد خروجی‌های دستور `dsolve` باید با تعداد معادلات موجود مطابقت داشته باشد، زیرا هر معادله منجر به یک تابع حل جداگانه می‌شود. این رویکرد در MATLAB امکان تحلیل دقیق و تحلیل سیستم‌های چندمعادله‌ای را فراهم می‌سازد و به کاربر اجازه می‌دهد پاسخ‌های سمبولیک هر متحول وابسته را به صورت جداگانه استخراج نماید (Arefin et al., 2020). این جا تمرکز ما به بررسی و حل معادلات تفاضلی معمولی (ODE) در محیط برنامه MATLAB است. معادلات ODE، بسته به شرایط و ساختار آن، ممکن است دارای حل تحلیلی باشند. به طور کلی، دو رویکرد اصلی برای حل این دسته از معادلات وجود دارد؛ حل تحلیلی (Analytical Solution) و حل عددی (Numerical Solution) است. در MATLAB، برای یافتن حل تحلیلی معادلات تفاضلی معمولی از دستور `dsolve` استفاده می‌شود. این دستور معادله را به صورت سمبولیک دریافت کرده و پاسخ



دقیق و تحلیلی آن را باز می‌گرداند. جهت تعریف مشتقات در محیط سمبولیک MATLAB، از دستور diff بهره گرفته می‌شود (Arefin et al., 2020; Petrás, 2011).

مثال ۱: حل معادله تفاضلی درجه ۱ در MATLAB

$$x \frac{dy}{dx} - 2y = 2x^5$$

```
Command Window
>> syms y(x)
eqn = x* diff(y,x) - 2*y == 2*x^5;
dsolve(eqn)

ans =

(2*x^5)/3 + C2*x^2
```

شکل ۹: خروجی برنامه MATLAB در حل معادله تفاضلی درجه ۱

مثال ۲: معادله تفاضلی درجه ۲ در MATLAB

$$y'' - 3y' + 2y = e^{-x}$$

Diff (y, x,2) - 3*diff (y, x) +

کد برنامه MATLAB:

2*y== exp(-x)

```
Command Window
>> syms y(x)
eqn = diff(y,x,2) - 3*diff(y,x) + 2*y == exp(-x);
dsolve(eqn)

ans =

exp(-x)/6 + C3*exp(x) + C4*exp(2*x)
```

شکل ۱۰: خروجی برنامه MATLAB در حل معادله تفاضلی درجه ۲

مثال: حل سیستم معادلات تفاضلی در MATLAB



$$\begin{cases} y' = 3y + 4z \\ z' = -4y + 3z \end{cases}$$

$$\begin{aligned} \text{eqn1} &= \text{diff}(y, x) = \\ &= 3 * y + 4 * z; && \text{MATLAB: برنامه} \\ \text{eqn2} &= \text{diff}(z, x) = \\ &= -4 * y + 3 * z; \end{aligned}$$

```
Command Window
>> syms y(x) z(x)
eqn1 = diff(y,x) == 3*y + 4*z;
eqn2 = diff(z,x) == -4*y + 3*z;
eqn = [eqn1, eqn2];
[y_ans, z_ans] = dsolve(eqn)

y_ans =
C6*cos(4*x)*exp(3*x) + C5*sin(4*x)*exp(3*x)

z_ans =
C5*cos(4*x)*exp(3*x) - C6*sin(4*x)*exp(3*x)
```

شکل ۱۱: حل سیستم معادلات تفاضلی در MATLAB

کاربرد MATLAB در علم احصائیه

احصائیه (Statistics) یکی از شاخه‌های اساسی و حیاتی ریاضیات کاربردی است که به تحلیل داده‌ها، بررسی توزیع‌ها و برآورد احتمال وقوع پدیده‌ها می‌پردازد. در این عرصه، برنامه MATLAB به‌عنوان یک ابزار محاسباتی پیشرفته، امکانات گسترده‌ای را برای تحلیل‌های احصائیه فراهم می‌سازد. MATLAB با بهره‌گیری از مجموعه‌ای کامل از توابع احصائیه، از جمله محاسبه اوسط (mean)، میانه (median) و انحراف معیار (standard deviation)، امکان تحلیل دقیق مجموعه‌های داده را فراهم می‌نماید. این توابع، نقش مهمی در بررسی ویژگی‌های مرکزی و پراکندگی داده‌ها ایفاء کرده و به‌خاص در انجام تست‌های فرضیه (hypothesis testing) و تحلیل نتایج تجربی در حوزه‌های مختلف علمی از اهمیت خاصی برخوردارند. با استفاده از جعبه‌ابزارهای تخصصی احصائیه در MATLAB (Statistics and Machine Learning Toolbox)، محققان قادر



خواهند بود مدل‌سازی‌های احصائیوی پیشرفته، تحلیل رگرسیون، تحلیل واریانس و شبیه‌سازی‌های آماری را نیز با دقت بالا انجام دهند. از این رو، MATLAB به‌عنوان یک محیط یکپارچه برای تحلیل احصائیوی داده‌ها، نقش بسزایی در مطالعات علمی - تحقیقی، صنعتی و آکادمیک ایفاء می‌کند (Chen, 2022; Pawade, 2023)

مزایای استفاده از MATLAB در تحقیق‌های ریاضی

برنامه MATLAB به‌عنوان یک ابزار قدرتمند محاسباتی، مزایای متعددی برای تحقیق در علوم ریاضی فراهم می‌سازد که آن را از روش‌های سنتی متمایز می‌سازد. نخست، MATLAB به دلیل برخورداری از محیطی مناسب و ساختار ساده، یادگیری و استفاده از آن برای کاربران در سطوح مختلف آسان است و همین ویژگی سبب شده تا تعداد زیادی محققین، از محصلان تا متخصصان، بتوانند به راحتی از آن بهره‌مند شوند. دوم؛ این برنامه توانایی بالایی در پروسس و تحلیل داده‌های بزرگ دارد و می‌تواند محاسبات عددی پیچیده را با دقت و سرعت بالا انجام دهد؛ قابلیت‌هایی که در بسیاری از تحلیل‌های ریاضی ضروری است. افزون بر آن، MATLAB ابزارهای متنوعی برای تجسم داده‌ها و نتایج در اختیار قرار می‌دهد، که این امر امکان ارائه بصری و قابل فهم یافته‌های تحقیق را فراهم می‌سازد و ارتباط مؤثرتر با مخاطب علمی را تقویت می‌کند. در نهایت، MATLAB یک محیط برنامه‌نویسی منعطف و توسعه‌پذیر در اختیار محققین قرار می‌دهد که از طریق آن می‌توانند الگوریتم‌های موجود را به دلخواه تغییر داده یا الگوریتم‌های جدیدی را متناسب با نیاز تحقیق طراحی و پیاده‌سازی نمایند. این ویژگی‌ها در مجموع، MATLAB را به ابزاری ممتاز برای انجام تحقیقات ریاضی در زمینه‌های مختلف تبدیل کرده‌اند در نهایت مزایای استفاده از این برنامه قرار ذیل است (Niazai et al., 2023).



پیاده‌سازی الگوریتم‌ها: برنامه MATLAB مجموعه زیادی از توابع و جعبه‌ابزارهای داخلی را فراهم می‌کند که به محققین این امکان را می‌دهد الگوریتم‌های ریاضی را به سرعت طراحی، آزمایش و ارزیابی نمایند. محیط برنامه‌نویسی منعطف: برنامه MATLAB بستری انعطاف پذیر برای توسعه الگوریتم‌ها ریاضی فراهم می‌سازد، به گونه‌ای که کاربران می‌توانند مدل‌های محاسباتی خود را مطابق با نیاز تحقیق تغییر دهند یا الگوریتم‌های جدیدی طراحی کنند.

قابلیت تحلیل و مدیریت: برنامه MATLAB توانایی بالایی در تحلیل مجموعه داده‌های بزرگ را دارند و محاسبات عددی پیچیده را به صورت مؤثر مدیریت می‌کند؛ این ویژگی آن را برای تحلیل‌های احصائیوی، الگوریتم‌های ریاضی، یادگیری ماشین و مدل‌سازی در حوزه‌های مختلف علمی مناسب می‌سازد.

ابزارهای تجسم‌سازی قدرتمند: برنامه MATLAB ابزارهای پیشرفته‌ای برای نمایش بصری داده‌ها و نتایج، بصورت گراف‌ها و چارت‌ها ارائه می‌دهند که انتقال مفاهیم پیچیده ریاضی را ساده‌تر و مؤثرتر می‌سازد.

قابلیت یکپارچگی با زبان‌های دیگر: برنامه MATLAB از زبان‌های برنامه‌نویسی مختلف مانند Python، C++ و Java پشتیبانی می‌کند و به راحتی در محیط‌های ترکیبی تحقیقاتی قابل استفاده است (Geete & Verma, 2023b; Pawade, 2023).

نقاط قوت برنامه MATLAB

۱. **قابلیت‌های چندگانه:** برنامه MATLAB هم‌زمان می‌تواند به عنوان یک ماشین حساب پیشرفته و نیز یک زبان برنامه‌نویسی تخصصی مورد استفاده قرار گیرد. این نرم‌افزار ترکیبی قوی از امکانات محاسباتی عددی، ترسیم چارت‌های گرافیکی و تحلیل داده را در اختیار کاربر قرار می‌دهد.



۲. **سهولت یادگیری:** محیط کاربرپسند و ساختار ساده MATLAB باعث می‌شود یادگیری آن، بطور خاص برای محصلان و کاربران تازه‌کار، سریع و آسان باشد.

۳. **امکان رفع خطاها:** برنامه MATLAB یک زبان تفسیری (Interpreted) است، به این معنا که کدها خط به خط اجرا می‌شوند. این ویژگی امکان شناسایی و رفع سریع خطاها را فراهم می‌سازد و پروسه اجراء و اصلاح کد را تسهیل می‌کند.

۴. **کارایی بالا در عملیات ماتریکسی:** برنامه MATLAB به‌طور خاص برای انجام محاسبات ماتریکسی و عددی طراحی شده و عملکرد بسیار مناسبی در این زمینه دارد. این ویژگی آن را به ابزاری قدرتمند در تحلیل‌های ریاضی و انجینری تبدیل کرده است.

۵. **پشتیبانی از برنامه‌نویسی شی‌گرا:** برنامه MATLAB از برخی مفاهیم برنامه‌نویسی شی‌گرا (Object-Oriented Programming) پشتیبانی می‌کند که به توسعه ساختار یافته و منظم برنامه‌ها کمک می‌کنند (Geete & Verma, 2023a).

نقاط ضعف برنامه MATLAB

۶. **عدم تطابق با زبان‌های برنامه‌نویسی عمومی:** MATLAB یک زبان عمومی و همه‌منظوره مانند C++ یا Python نیست و عمدتاً برای محاسبات علمی و انجینری طراحی شده است. از این رو، استفاده از آن در پروژه‌های غیرعلمی با محدودیت‌هایی مواجه است.

۷. **کاربرد محدود در حوزه‌های غیرعلمی:** طراحی برنامه MATLAB به‌طور خاص برای کاربردهای علمی، و ریاضی بوده و در حوزه‌هایی مانند طراحی وب، برنامه‌های کاربردی عمومی یا توسعه نرم‌افزارهای تجاری، مناسب و کارآمد نیست.



۸. **سرعت پایین تر نسبت به زبان های کامپایلری:** به دلیل ماهیت تفسیری برنامه MATLAB، سرعت اجرای برنامه ها معمولاً کمتر از زبان های کامپایلری مانند ++C یا Fortran است، به خاص در پروژه هایی که نیاز به پردازش سنگین و بهینه سازی دارند.

۹. **وابستگی به محیط خاص MATLAB:** بسیاری از دستورات و قابلیت های MATLAB تنها در محیط خود این برنامه قابل استفاده اند و قابل انتقال مستقیم به زبان های دیگر نیستند. این موضوع یک محدودیت در توسعه بین پلتفرمی محسوب می شود.

۱۰. **عدم تطابق دستورات با سایر زبان ها:** ساختار و نحوه نگارش دستورات برنامه MATLAB با بسیاری از زبان های برنامه نویسی دیگر تفاوت دارد، بنابراین مهاجرت از این برنامه به زبان های دیگر ممکن است با چالش هایی همراه باشد.

۱۱. **محدودیت در استفاده از کد در محیط های دیگر:** اغلب کدهای نوشته شده در MATLAB تنها در همین محیط قابل اجرا هستند و به راحتی نمی توان آن ها را در سایر سیستم ها یا زبان های برنامه نویسی به کار گرفت، مگر با استفاده از ابزارهای جانبی یا تبدیل کد برنامه (Geete & Verma, 2023b).

نتیجه گیری

برنامه MATLAB، به عنوان یکی از ابزارهای پیشرفته محاسباتی، نقش مهمی در حل و تحلیل مسائل ریاضی ایفاء می کند. این برنامه با ترکیب محاسبات عددی و سمبولیک، محیطی توانمند برای مدل سازی الگوریتم ها، تحلیل احصائیوی، محاسبه مشتق، انتیگرال و حل سیستم های معادلات را فراهم می سازد. محیط کاربری ساده و آسان، قابلیت تجسم سازی و ادغام با سایر زبان های برنامه نویسی از ویژگی های برجسته آن است. یافته های این تحقیق نشان می دهد که برنامه



MATLAB در حل محاسبات ریاضی کاربرد زیادی دارد. تسلط بر این برنامه برای محصلان و محققان ضروری بوده و استفاده گسترده از آن می تواند دقت جواب در حل محاسبات پیچیده را افزایش دهد. اضافه نمودن آموزش برنامه MATLAB در نصاب درسی برنامه های علمی، گامی مهم در ارتقای کیفیت تحقیق های علمی به شمار می رود.



References

- Arefin, M. A., Gain, B., Karim, R., & Hossain, S. (2020). A comparative exploration on different numerical methods for solving ordinary differential equations. *J. Mech. Cont. & Math. Sci*, 15(12), 1-11.
- Chen, X. (2022). Application of MATLAB in College Students Mathematical Modeling Competition. 2nd International Conference on Internet, Education and Information Technology (IEIT 2022),
- Geete, N., & Verma, S. (2023a). MATLAB Application for Mathematics Students. *International Journal of Mathematics and Computational Research*, 11, 3138-3141.
- Geete, N., & Verma, S. (2023b). MATLAB Applications for Mathematics Students.
- Kazimovich, Z. M., & Guvercin, S. (2012). Applications of symbolic computation in MATLAB. *International Journal of Computer Applications*, 41(8), 1-5.
- Lopez, C. (2014). *MATLAB Symbolic Algebra and Calculus Tools*. Apress.
- Moler, C., & Little, J. (2020). A history of MATLAB. *Proceedings of the ACM on Programming Languages*, 4(HOPL), 1-67.
- Niazai, S., Rahimzai, A. A., & Atifnigar, H. (2023). Applications of MATLAB in natural sciences: a comprehensive review. *Eur J Theoret Appl Sci*, 1(5), 1006-1015.
- Norddin, N. I., Fadhilah, N. H., Shah, M. A. A., Yusoff, N. H. M., & Ahmad, N. I. S. (2025). Nonlinear Equation Solver with MATLAB Graphical User Interface. *Journal of Computing Research and Innovation*, 10(1), 157-167.
- Overman, E. (2011). A matlab tutorial. *Dept. of Mathematics, Ohio State Univ., Columbus, OH*, 180.
- Pawade, I. (2023). The application of MATLAB in mathematics: an overview. *International Journal For Multidisciplinary Research*, 5(5).
- Petrás, I. (2011). *Fractional derivatives, fractional integrals, and fractional differential equations in Matlab*. IntechOpen London, UK.
- Thabet, K., & Al-Ghuribi, S. (2012). Matrix multiplication algorithms. *International Journal of Computer Science and Network Security (IJCSNS)*, 12(2), 74.
- Xue, D. (2020). *Linear Algebra and Matrix Computations with MATLAB®*. Walter de Gruyter GmbH & Co KG.



A Review of MATLAB Applications in Numerical Mathematical Computations

Ahmad Wali Noori^{1*}, Farhad Hakimi², Mohammad Yazdan Parast³,
Mathematics Department, Education Faculty, Farah University

Email: ahmadwalinoori@farau.edu.af

Abstract

In the current era, the complexity of scientific and research problems has significantly increased the demand for precise, flexible, and efficient computational tools. MATLAB, as an advanced computational program and a high-level programming language, has attained a distinguished status among such tools. This study aims to examine the key applications of MATLAB in solving numerical mathematical computations related to linear algebra, linear and nonlinear equations, derivatives, integrals, differential equations, matrices, and statistical analyses. The theoretical framework of this research is based on a library-based method, wherein credible scientific sources have been reviewed to evaluate MATLAB's capabilities in addressing complex computational problems. The findings indicate that MATLAB, with its specialized toolboxes, predefined functions, advanced visualization capabilities, and support for object-oriented programming languages, provides a robust platform for conducting sophisticated mathematical analyses, teaching and implementing algorithms, and executing interdisciplinary projects. Indeed, MATLAB can be regarded as a reliable, flexible, and transformative tool in solving complex mathematical computations.

Keywords: Mathematical Problem, MATLAB Applications, Numerical Computations